This listing of claims will replace all prior versions, and listings, of claims in the application.

Listing of Claims:

1. (Currently amended) A ~~computer-implemented method of processing runtime functions,~~ computer-readable storage medium having stored thereon computer-executable instructions for performing a method of processing runtime functions, the method comprising:

compiling source code files to produce ~~executable code that~~ a plurality of object files, each of which is marked with an identifier for indicating that ~~the~~ executable code generated from each of the plurality of object files supports ~~comprises an object file containing a list of valid target addresses for use in implementing~~ runtime protection;

concluding that a first object file has no valid target addresses pertaining to runtime functionality, upon detecting: a) the presence of the identifier in the first object file, and b) the absence of a runtime section of code in the first object file;

concluding that a second object file has a list of valid target addresses pertaining to runtime functionality, upon detecting: a) that the identifier in the second object file is present and has been asserted, and b) that a runtime section of code is present in the second object file;

storing in a table of the second object file, the list of valid target addresses as a reference list of valid target addresses for implementing a runtime function associated with the runtime section of code contained in the second object file;

receiving a call to ~~a~~ the runtime function, upon execution of the executable code of the second object file;

determining associated data from the call to the runtime function;

determining a target address from the associated data;

comparing the target address with the reference list of valid target addresses stored in the table;

if the target address is found on the reference list of valid target addresses then executing the runtime function; and

if the target address is not found on the reference list of valid target addresses then terminating execution of the runtime function.

2. (Original) The method of claim 1, wherein the step of determining the associated data comprises accessing data in a data structure connected with the runtime function and calculating the associated data based on the accessed data.

3-5 (Canceled).

6. (Previously presented) The method of claim 1 comprising the step of generating the reference list of valid target addresses during execution of a previous runtime function.

7-9. (Canceled)

10. (Withdrawn) A computer-readable storage medium having stored thereon computer-executable instructions for performing a method of processing runtime functions, the method comprising:

receiving a call to a runtime function;

determining associated data from the call to the runtime function;

deriving a security cookie by XORing a secret value with each of the values retrieved from a jmp_buf buffer, the retrieved values precluding a first security cookie that has been stored previously in the jmp_buf buffer;

comparing the derived security cookie against the first security cookie; and

if the derived security cookie matches the first security cookie then executing the runtime function; and

if the derived security cookie does not match the first security cookie then terminating execution of the runtime function.

11. (Withdrawn) The computer-readable storage medium of claim 10, wherein the step of determining the associated data comprises accessing data in a data structure connected with the runtime function and calculating the associated data based on the accessed data.

12-14. (Canceled)

15. (Withdrawn) The computer-readable storage medium of claim 10 wherein the first security cookie is derived during execution of a previous runtime function.

16-18. (Canceled)

19. (Currently amended) A computer system that includes a computer-readable storage medium having stored thereon program modules for processing runtime functions, the program modules comprising:

a compiler that compiles source code files to produce ~~an executable that~~ a plurality of object files, wherein each of the plurality of files is produced by the compiler ~~is marked~~ with an identifier ~~indicating that the executable comprises an~~ that operates as a marker which when placed in a set condition provides an indication that the object file ~~containing~~ contains a list of valid target addresses for use in implementing runtime protection;

a processor that receives a call to a runtime function; and

a dispatcher system that determines associated data from the call to the runtime function, determines a target address from the associated data, and if the target address is found on the list of valid target addresses then executes the runtime function.

20. (Original) The system of claim 19, wherein the dispatcher system comprises a module to access data in a data structure connected with the runtime function and calculate the associated data based on the accessed data.

21-22. (Canceled)

23. (Currently amended) The system of claim 19, ~~further comprising a~~ wherein the compiler ~~that~~ generates the list of valid target addresses.

24-27. (Canceled)

28. (Previously Presented) The method of claim 1 wherein the step of storing in the table comprises storing in a caller provided location during execution of a previous runtime function.

29-36. (Canceled)

37. (Previously Presented) The method of claim 1, further comprising:
determining if at least a portion of the associated data is valid; and
preventing execution of the runtime function if the associated data is not valid.

38. (Previously presented) The method of claim 37, wherein the step of determining if the associated data is valid comprises retrieving a security cookie from the associated data and comparing the retrieved security cookie to a list of valid security cookies.

39. (Previously Presented) The method of claim 37, further comprising determining and storing a predetermined calculated value based on at least a portion of the associated data, prior to receiving the call to the runtime function.

40. (Previously Presented) The method of claim 39, wherein determining if the associated data is valid comprises comparing the predetermined calculated value to another calculated value based on the associated data.

41. (Withdrawn) The computer readable medium of claim 10, having further computer-executable instructions for determining if at least a portion of the associated data is valid, and preventing execution of the runtime function if the associated data is not valid.

42. (Canceled)

43. (Withdrawn) The computer-readable medium of claim 41, having further computer-executable instructions for determining and storing a predetermined calculated value based on at least a portion of the associated data, prior to receiving the call to the runtime function.

44. (Withdrawn) The computer-readable medium of claim 43, wherein determining if the associated data is valid comprises comparing the predetermined calculated value to another calculated value based on the associated data.

45. (Previously Presented) The system of claim 19, wherein the dispatcher system comprises modules to determine if at least a portion of the associated data is valid and prevent execution of the runtime function if the associated data is not valid.

46. (Previously Presented) The system of claim 45, further comprising a storage device that stores the list of valid target addresses.

47. (Previously Presented) The system of claim 45, wherein the dispatcher system determines if the associated data is valid by retrieving a security cookie from the associated data and comparing the retrieved security cookie to a list of valid security cookies.

48. (Previously Presented) The system of claim 45, wherein the processor determines and stores a predetermined calculated value based on at least a portion of the associated data, prior to receiving the call to the runtime function.

49. (Previously Presented) The system of claim 48, wherein the dispatcher system determines if the associated data is valid by comparing the predetermined calculated value to another calculated value based on the associated data.

50. (Currently amended) The method of claim 1, wherein the identifier is an identifier bit that is operable to be set for indicating assertion of the identifier, ~~that the executable code comprises the object file containing the list of valid target addresses for use in implementing runtime protection.~~

51. (Currently amended) The method of claim 1, wherein the table is a [[.]]setjmp table and the call to the runtime function is a longjmp.